An Interop Publication

ConneXions®

The Interoperability Report

June 1994 Volume 8, No. 6

ConneXions — The Interoperability Report tracks current and emerging standards and technologies within the computer and communications industry.

In this issue:

ConneXions is published monthly by Interop Company, a division of ZD Expos, 303 Vintage Park Drive, Foster City, California, 94404–1138, USA.
Phone: +1 (415) 578-6900
Fax: +1 (415) 525-0194
E-mail: connexions@interop.com

ISSN 0894–5926

From the Editor

When we launched *ConneXions* seven years ago, we had two ideas for its contents. First, we would cover new and emerging technologies with tutorial articles and standardization updates. Second, we would explain existing ("old") technologies and protocols and show how they could be used to build modern internets. Over the years there has been a mix of both types of articles in this journal, perhaps with a slight bias towards the "new" at the expense of the "old." This month, we begin a new series of articles under the heading "Back to Basics." From time to time, we will bring you articles about well-established techniques and technologies, the fundamentals of networking. Our first installment is a tutorial on the Internet Transport Layer (TCP and UDP) and is adapted from a forthcoming book on network management by Keith McCloghrie and Marshall Rose.

Networking continues to be a growth industry world wide. This month, we look at DANTE and EuropaNET, an effort to establish a pan-European research network. The article is by Josefien Bersee.

If you want access to most of the tools on the Internet (*FTP, Telnet, WWW, Gopher,* etc.), you need IP connectivity in one form or another. The simplest and cheapest way to gain such access is to use SLIP or PPP with a dialup connection from your workstation (PC) to your service provider. Billy Barron compares a number of public domain SLIP and PPP drivers starting on page 16.

As more and more businesses join the Internet, the question of electronic commerce becomes important. *NetCash* is a framework for electronic currency being developed at USC-ISI. Gennady Medvinsky and Cliff Neuman give an overview of the system.

The *NetWorld+Interop 94 World Tour* has begun. Last month in Las Vegas over 60,000 visitors attended the inaugural event, more than 7,000 of them taking part in the tutorials, workshops or conference sessions. Visitors were able to see the first ever "Cyberstation" which broadcast sound, text and images live to the Internet. The exhibition featured the traditional InteropNet interconnecting the 600+ exhibitors, constructed with more than 300 miles of cable and 4,000 pieces of donated equipment. The World Tour continues in Berlin (June 6–10), Tokyo (July 25–29), Atlanta (September 12–16), and finally Paris (October 24–28). For more information about these events, call 1-800-INTEROP, 1-415-578-6900 or check out our *WWW* and *Gopher* servers on `programs.interop.com`.

Finally, another reminder that we would like to hear from you. Send your suggestions, comments and questions to: *ConneXions,* 303 Vintage Park Drive, Foster City, CA 94404–1138, USA. You can also reach us by fax: 415-525-0194 or e-mail: `connexions@interop.com`.

Back to Basics: The Internet Transport Layer

by
Keith McCloghrie, Cisco Systems and
Marshall T. Rose, Dover Beach Consulting

Introduction

The transport layer is responsible for providing data transfer between end-systems to the level of reliability desired by the application. That is, the transport layer provides *end-to-end service.*

In theory, the end-to-end needs of different applications can vary tremendously. In practice, however, there are really only two widely-used service paradigms:

1. *Reliable:* in which the service offered is a "virtual pipeline":

- *Stream-oriented:* rather than dealing in packet exchanges, the end-to-end service provides a sequence of octets, termed a stream, to the application.
- *Full-duplex:* the stream provided by the end-to-end service is bi-directional in nature.
- *Connection-oriented:* before the stream can be used, a virtual connection is established between the two applications.
- *Application layer addressing:* an application needs a means of identifying its peer on the remote system to which the stream should be connected.
- *In-sequence delivery:* the end-to-end service guarantees that user-data is delivered in the same order in which it was sent.
- *User-data integrity:* the end-to-end service guarantees that any user-data delivered has not been corrupted during network transmission.
- *Graceful release:* because user-data may be buffered both at the hosts and in the network, the end-to-end service will make sure that *all* of the data sent by the user is successfully transmitted before the stream is released.

Note that these are general guidelines, and not fixed. In particular, the OSI CO-mode transport service, whilst offering a reliable transport paradigm, uses a packet-oriented (rather than stream-oriented) user-data paradigm, and has no graceful release mechanism (the functionality of which resides at the layer above). Regardless, the remaining characteristics are core to the concept of a reliable transport service.

2. *Unreliable:* in which the service offered is virtually identical to that of the Internet datagram service. The only added features are:

- Application layer addressing; and,
- User-data integrity.

It shouldn't be surprising that the reliable service paradigm corresponds closely to a connection-oriented transport service, whilst the unreliable service paradigm is similar to a connectionless-mode transport service.

The Internet suite of protocols provides two different transport protocols to meet these vastly different needs. Since both protocols use identical mechanisms to achieve application layer addressing and user-data integrity, the simpler protocol is described first.

UDP

The *User Datagram Protocol* (UDP) is the connectionless-mode transport protocol in the Internet suite. As UDP is a transport layer protocol, for delivery, it uses the services of IP. If the *protocol* field of an IP datagram has the value 17 (decimal), the user-data contained in the datagram is a UDP packet:

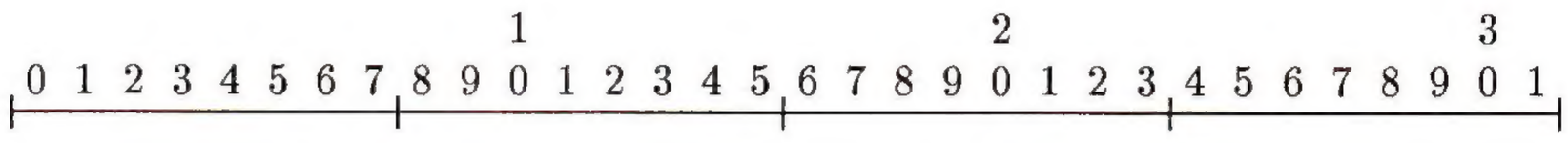


Figure 1: A UDP Packet

The meaning of these fields is straightforward:

- *Source/destination port:* identifies an application running at the corresponding IP address.
- *Length:* the length of the UDP packet (header and user-data), measured in octets.
- *Checksum:* a one's-complement arithmetic sum, computed over a *pseudo-header* and the entire UDP packet.
- *User-data:* zero or more octets of data from the upper-layer protocol. (Note that it is an artifact of the convention used in producing the figure above that this field appears to be a multiple of 4 octets in length. No such requirement is made by UDP.)

The uses of these fields are now explained.

Application Layer Addressing

To achieve application layer addressing, UDP manages 16-bit unsigned integer quantities, termed *ports*. Port numbers less than 512 are assigned by the *Internet Assigned Numbers Authority* (IANA).

These are termed *well-known ports*. In those cases when a service might be available over both TCP and UDP, the IANA assigns the same port number to that service for both protocols.

On Berkeley UNIX, port numbers less than 1024 are reserved for privileged processes (an easily spoofed but, in some environments, useful security mechanism).

The combination of an IP address and a port number is termed an internet *socket* which uniquely identifies an application-entity running in an internet.

Of course, the notion of application layer addressing is just another example of the multiplexing operation of protocols:

- At the interface layer, each medium usually distinguishes between clients (entities at the internet layer) by using different values in a *type* field (e.g., Ethernet uses a value of 0x0800 to indicate IP);
- At the internet layer, IP distinguishes between clients (entities at the transport layer) by using different values in a *protocol* field (e.g., IP uses a value of 17 to indicate UDP); and,

continued on next page

The Internet Transport Layer *(continued)*

- At the transport layer, TCP and UDP distinguish between clients (entities at the application layer) by using different values in a *port* field (e.g., UDP uses a value of 161 (decimal) to indicate SNMP).

The Assigned Numbers RFC lists the complete set of protocol numbers used at all layers in the Internet suite of protocols.

User-Data integrity

To achieve both user-data integrity and modest protection against misbehavior at the layers below, UDP calculates a *pseudo-header* which is conceptually prefixed to the UDP packet. The checksum algorithm is then run over a block that looks like this:

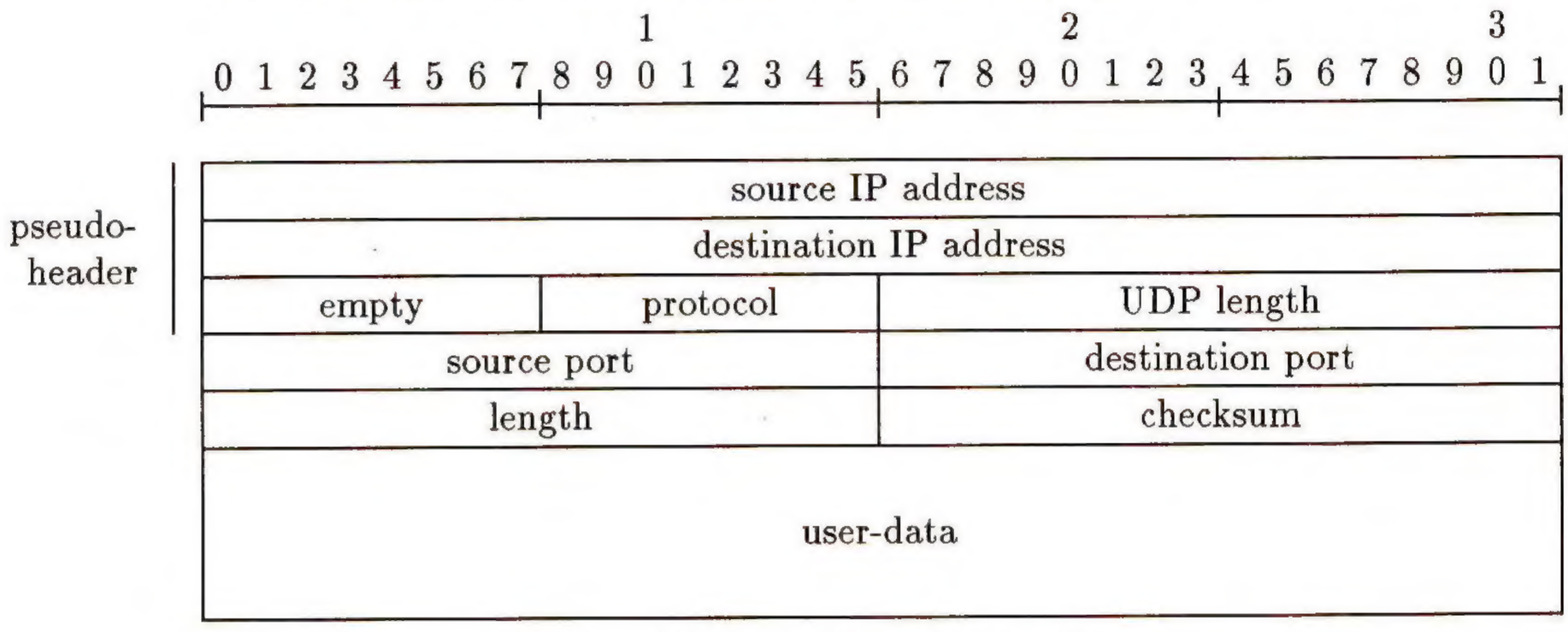


Figure 2: Pseudo-header

The fields of the pseudo-header are relatively self-explanatory: the *source* and *destination* fields are taken from the IP packet, the *empty* field is simply a zero-valued octet, the *protocol* field is the value used by IP to identify UDP (17 decimal), and the *UDP length* field is the length of the UDP packet. TCP also uses this 96-bit pseudo-header in its checksum calculation when achieving user-data integrity.

TCP

The *Transmission Control Protocol* (TCP) is the connection-oriented transport protocol in the Internet suite. As TCP is a connection-oriented transport protocol, it goes through three distinct phases: connection establishment, data transfer, and connection release. To keep track of a particular connection, each TCP entity maintains a *Transmission Control Block* (TCB). This is created during connection establishment, modified throughout the life of the connection, and then deleted when the connection is released.

TCP is best described as a finite state machine, which starts in the CLOSED state. As *events* occur (either activity from a user of TCP or from the network), the TCP entity performs some *action* and then enters a new state. The TCP state diagram is presented in Figure 3. (It is suggested that the reader study the text before examining the figure.)

Connection establishment

A connection enters the LISTEN state when an application tells TCP that it is willing to accept connections for a particular port number. This is termed a *passive open.*

Sometime later, another application tells TCP that it wishes to establish a connection to an IP address and port number which corresponds to the application which is listening. This is termed an *active open.* (It is possible for two application entities to simultaneously issue active opens for each other. In this case, a single TCP connection is established.)

When two TCP entities communicate, the exchanged units of data are termed *segments*. The format of a segment is presented later on. Segments are interpreted relative to a *connection*. In TCP, a connection is defined as the pairing of the two internet sockets. This 96-bit quantity (source IP address and TCP port, destination IP address and TCP port) uniquely identifies the connection in an internet.

When an active open is attempted, the originating TCP entity computes an *initial sequence number,* which is a "starting number" for this direction of the new connection. The sequence number must be chosen carefully so that segments from older instances of this connection, which might be floating around the network, won't cause confusion with this new connection. A SYN (synchronize) segment is then sent to the destination TCP entity. Upon receiving this segment, the destination TCP entity checks to see that an application is listening on the destination TCP port. If not, the connection is aborted by sending a RST (reset) segment. (In the interest of simplicity, Figure 3 doesn't show this transition, or any transition, involving an RST segment.) Otherwise, the destination TCP entity computes a sequence number for its direction, and sends this back in a SYN/ACK (synchronize/acknowledge) segment which acknowledges the sequence number for the originating TCP entity.

Upon receiving this segment, the original TCP entity makes sure that its sequence number was acknowledged and, if all is well, sends an ACK segment back to acknowledge the sequence number for the destination TCP entity.

This protocol interaction is termed a *three-way handshake.* Once the three-way handshake has been successfully concluded, the connection enters the data transfer phase.

Data transfer

In the data transfer phase, user-data is sent as a sequence of octets, each of which is numbered, starting with the initial sequence number.

Each segment specifies a *window size* (in octets) which may be sent in the other direction before an acknowledgement is returned. Each segment sent by a TCP entity contains an implicit acknowledgement of all octets contiguously received thus far. Precisely stated, the acknowledgement field indicates the number of the *next* octet that is expected by a TCP entity.

This windowing strategy allows the TCP entities to achieve a *pipelining effect* in the network, while at the same time providing a flow control mechanism. The pipelining effect increases throughput by keeping more data in the network, whilst the flow control mechanism prevents either TCP entity from overrunning the connection resources (such as buffers for user-data) of the other.

The disadvantage of this approach is that if segments are re-ordered, this information cannot be conveyed in an acknowledgement. For example, if two segments are sent, and the first one is delayed, the receiving TCP entity cannot acknowledge the second segment until it receives the first.

Retransmission

The discussion thus far has not considered loss or corruption of segments. Each time a TCP entity sends a segment, it starts a retransmission timer. At some time in the future, one of two events will happen first: either an acknowledgement for the segment will be received, and the timer can be stopped; or, the timer will expire. In this latter case, the TCP entity *retransmits* the segment and restarts the timer.

continued on next page

The Internet Transport Layer *(continued)*

Retransmission continues some number of times until eventually the TCP entity gives up and declares the transport connection to be aborted. That is, TCP emulates reliability through retransmission. The trick, of course, is knowing *when* to retransmit. If data is lost or corrupted in the network and the sending transport entity retransmits too slowly, then throughput suffers. If data is delayed or discarded due to congestion in the network and the transport entity retransmits too quickly then it merely adds to the congestion and throughput gets even worse!

The reader should appreciate that because of the service offered by IP; a TCP entity cannot distinguish between lossy or congested networks. Hence, TCP uses one of several adaptive algorithms to predict the latency characteristics of the network, which may fluctuate considerably because of other traffic.

The retransmission timeout usually varies for each segment, based on the recent history of latency and loss exhibited by the network. Work reported in [9, 10] suggests some novel, common sense insights into this problem.

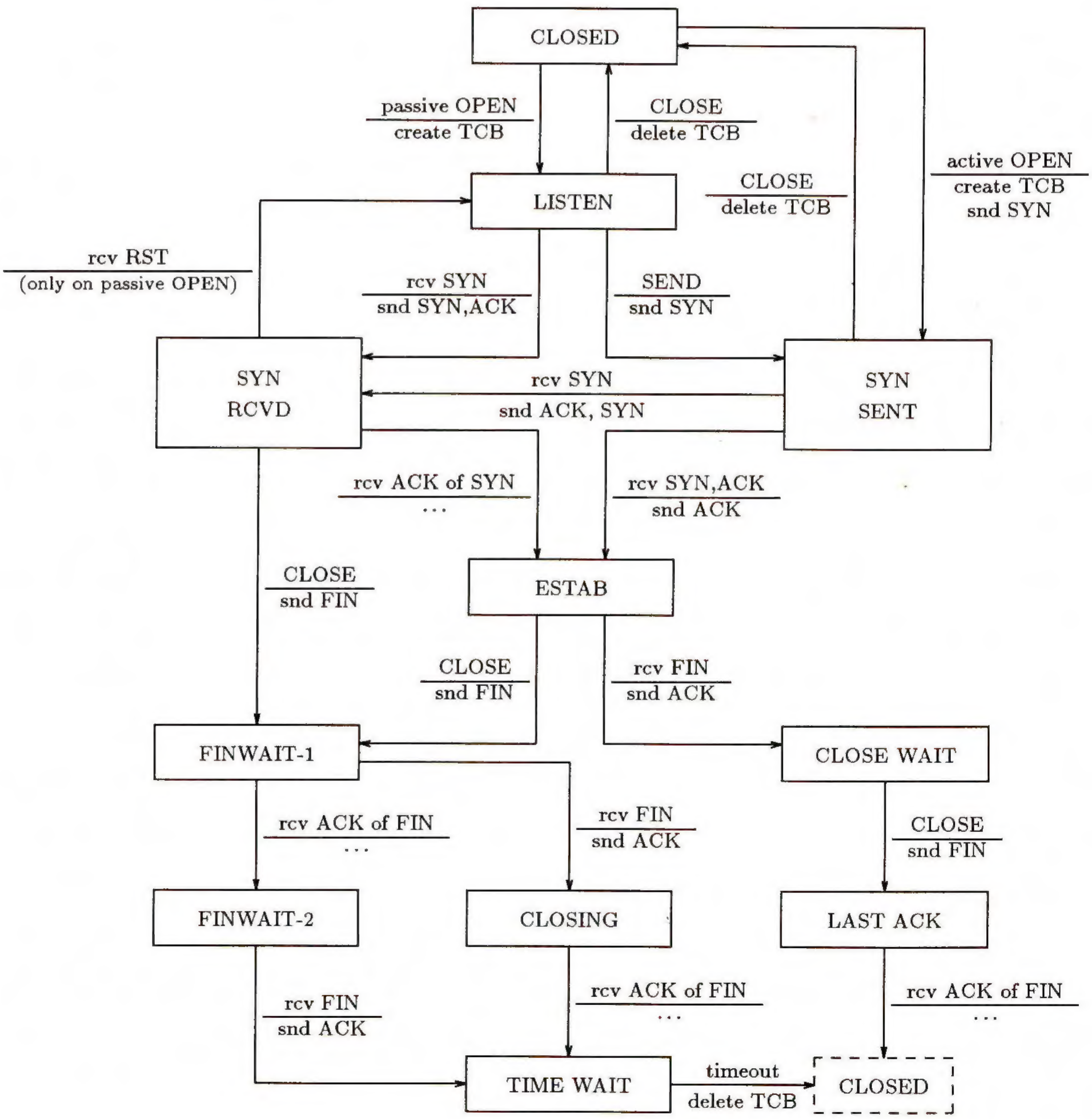


Figure 3: TCP State Diagram

As might be expected, acknowledgements and retransmission interact with the window strategy. Once again, suppose two segments are sent, and the first segment is lost. The receiving TCP entity cannot acknowledge the second segment. The retransmission timer expires for the sending TCP entity. It must now decide whether to retransmit the first segment or both segments. If it retransmits both segments, then it is "guessing" that both segments were lost. If this isn't the case, then network bandwidth is being wasted. Otherwise, if it retransmits the first segment only, it must wait for an acknowledgement to see if the second segment also needs to be retransmitted. If not, it has reduced its sending throughput by waiting for a roundtrip transaction in the network.

Queued delivery

In addition to trying to optimize network traffic, a TCP entity may try to reduce the overhead of communicating with local application entities. This is usually achieved by buffering user-data in the TCP entity, both as it is received from the local application, in order to efficiently use the network, and also as user-data is received from the network, in order to efficiently communicate with the local application. Because of this, an application might need a mechanism for ensuring that all data it has previously sent has been received.

This is accomplished using a PSH (push) function. When sending, an application may indicate that data previously sent should be pushed. The local TCP entity sets a PSH bit in the next new segment it sends. Upon receiving such a segment, the remote TCP entity knows that it should push user-data up to its own application.

Although the push function must be present in each TCP implementation, few implementations of applications actually use this functionality. This is because most TCP entity implementations will periodically push any queued data towards the destination. Further, it should be noted that there are no semantics associated with the push function. It is simply a way of telling TCP to deliver all data previously sent to the remote application. On the remote end, the application will see only the user-data and won't receive any explicit indication of the push function having been invoked. Experience has shown that the push function is largely an internal matter: application protocols should be designed so that the push function isn't used.

Urgent data

Finally, TCP supports the concept of *urgent data.* The semantics of urgent data are application-specific. What TCP does is to indicate where urgent data ends in the stream. The receiving application, upon being notified that urgent data is present in the stream, can quickly read from the stream until the urgent data is exhausted.

Connection release

When an application indicates that it has finished sending on the connection, the local TCP entity will send all outstanding data, setting the FIN (finish) indication in the last segment to indicate that it is finished sending new data.

Upon receiving this indication, the remote TCP entity will send an ACK for the FIN , and will inform (using a local mechanism) the application. When that application indicates that it too has no more data to send, a FIN is generated in this direction also. When all data in transit and the segments containing the FINs have been acknowledged, the two TCP entities declare the connection released. In order to ensure that old, duplicate packets don't interfere with new connections being established between the two application entities, one of the TCPs will delay releasing the connection by twice the maximum segment lifetime.

continued on next page

The Internet Transport Layer *(continued)*

Instead of requesting a graceful release, an application may determine that it wishes to immediately abort the connection. In this case, the local TCP entity generates a RST (reset) segment, and the connection is immediately released. Any data in transit is lost.

Segment format

When TCP wishes to send a segment, it uses the services of IP. If the *protocol* field of an IP datagram has the value 6 (decimal), the user-data contained in the datagram is a TCP segment:

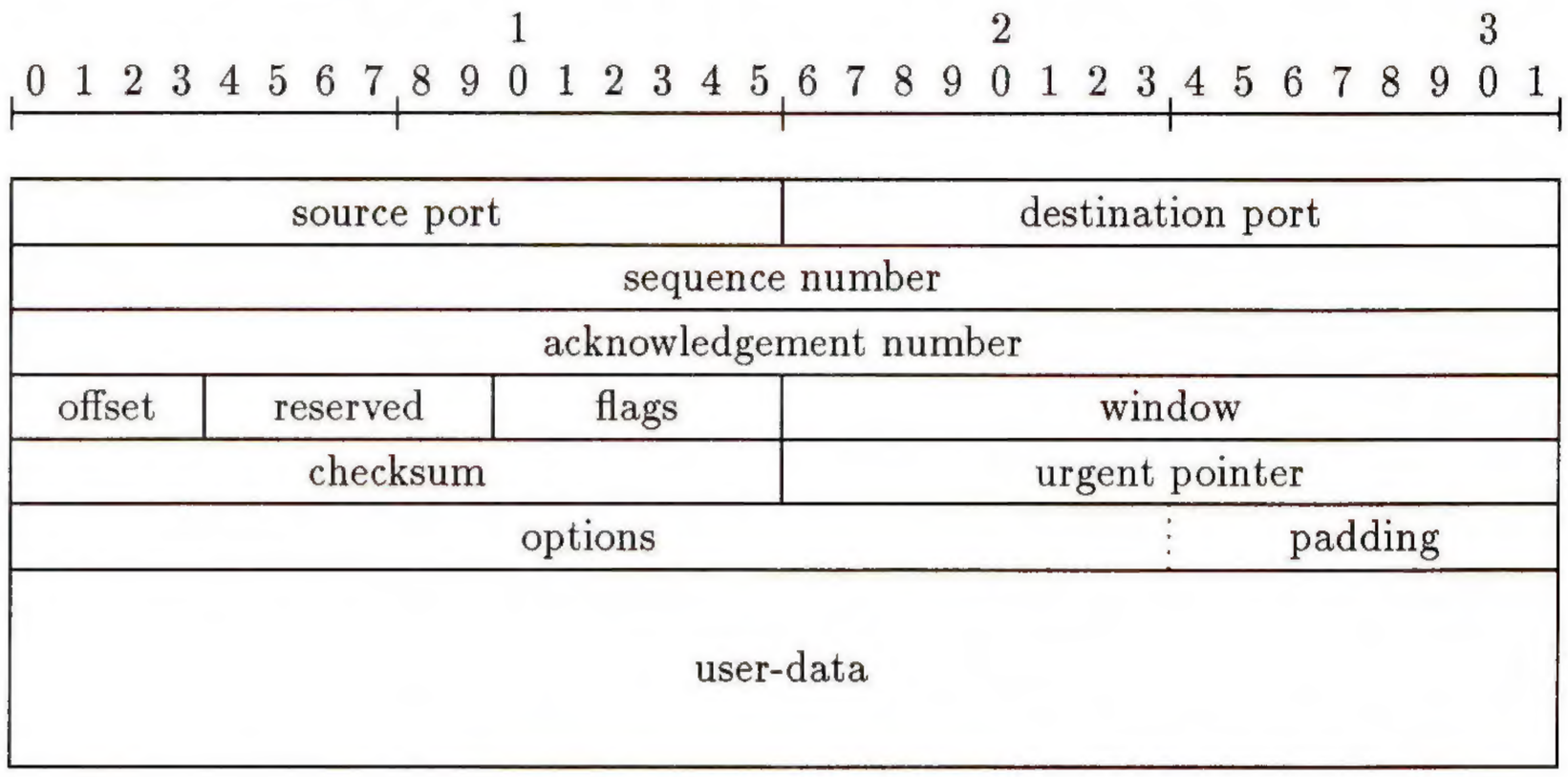


Figure 4: The TCP segment

The meaning of these fields is straightforward:

- *Source / destination port:* identifies an application running at the corresponding IP address.
- *Sequence number:* the number of the first octet of user-data in this segment.
- *Acknowledgement number:* if the ACK bit is set in the flags field, then this field indicates the next sequence number that the TCP entity is expecting to receive.
- *Offset:* the length of the TCP segment in 32-bit words (the minimum allowed value is 5).
- *Flags:* control bits indicating special functions of this segment.
- *Window:* the number of octets of user-data (starting with the octet indicated in the acknowledgement field), which the TCP entity is willing to accept.
- *Checksum:* a one's-complement arithmetic sum, computed over a pseudo-header and the entire TCP segment, as discussed earlier.
- *Urgent pointer:* if the URG bit is set in the flags field, then this field, when added to the sequence number field, indicates the first octet of non-urgent data.
- *Options:* a collection of zero or more options.
- *Padding:* zero to three octets used to pad the segment header to a 32-bit boundary.
- *User-data:* zero or more octets of data from the upper-layer protocol. (Note that it is an artifact of the convention used in producing the figure above that this field appears to be a multiple of 4 octets in length. No such requirement is made by TCP.)

References

[1] J. Reynolds, J. Postel, "Assigned Numbers," RFC 1340, July 1992.

[2] J. Postel, "Internet Protocol," RFC 791, September 1981.

[3] J. Postel, "User Datagram Protocol," RFC 768, August 1980.

[4] Jon Postel, editor, "Transmission Control Protocol," RFC 793, September 1981.

[5] Jain, Raj, "Divergence of Timeout Algorithms for Packet Retransmissions," Proceedings of the Fifth Annual International Phoenix Conference on Computers and Communications, 1986.

[6] Jain, Raj, "A Timeout-Based Congestion Control Scheme for Window Flow-Controlled Networks," in IEEE *Journal on Selected Areas in Communications,* October 1986.

[7] Jain, Raj, Ramakrishnan, K. K., and Chiu, Dah-Ming, "Congestion Avoidance in Computer Networks With a Connectionless Network Layer," Digital Equipment Corporation Technical Report, DEC-TR-506, August 1987.

[8] Zhang, Lixia, "Why TCP Timers Don't Work Well," Proceedings of SIGCOMM 1986, ACM Press.

[9] Partridge, Craig and Karn, Phil, "Improving Round-Trip Time Estimates in Reliable Transport Protocols," in Proceedings of SIGCOMM 1987, ACM Press.

[10] Jacobson, Van, "Congestion Avoidance and Control," in Proceedings of SIGCOMM 1988, ACM Press.

[11] Partridge, Craig, "Improving Your TCP: Look at the Timers," *ConneXions,* Volume 1, No. 3, July 1987.

[12] Partridge, Craig, "Improving Your TCP: Handling Source Quench," *ConneXions,* Volume 1, No. 7, November 1987.

[13] Partridge, Craig, "Improving Your TCP: Look under the hood!" *ConneXions,* Volume 2, No. 6, June 1988.

[14] Karn, Phil, "Improving Your TCP: 'Karn's Algorithm'," *ConneXions,* Volume 2, No. 10, October 1988.

[15] Partridge, Craig, "Improving Your TCP: Tuning the Checksum routine" *ConneXions,* Volume 2, No. 11, November 1988.

[Ed.: This article is adopted from *How to Manage Your Network using SNMP,* by Keith McCloghrie and Marshall Rose, ISBN 0-13-141517-4, to be published by Prentice-Hall in September 1994. Used with permission].

KEITH McCLOGHRIE is a Technical Leader at Cisco Systems, Inc. where he is involved in ATM development and in Network Management. He is a member of the IETF's Network Management Directorate and has been an active member of the SNMP working group since its inception, co-authoring many SNMPv1, SNMPv2 and MIB specifications. He is also a member of the ATM Forum's Technical Committee, contributing to the development of the UNI 2.0 and 3.0 specifications, and is currently chair of the LAN Emulation Working Group. He gained his B.Sc. in Mathematics from Manchester University in England.E-mail: `kzm@cisco.com.`

MARSHALL T. ROSE is Principal at Dover Beach Consulting, Inc., a California-based computer-communications consultancy. He spends half of his time working with clients, and the other half involved in self-supported, openly-available projects, as a theorist, implementor, and agent provocateur. He is the author of several books on computer networking: *The Open Book, The Little Black Book, The Internet Message* and *The Simple Book* (two editions)—all published by Prentice Hall. His subscriptions to *The Atlantic, Rolling Stone Magazine,* and *Wired!* are in good standing. He can be reached on the Internet as: `mrose@dbc.mtview.ca.us`

Profile: DANTE and EuropaNET

by Josefien Bersee, DANTE

Introduction

For many years the introduction of cross-border network services between European countries was the result of independent, bilateral agreements between pairs of national organisations, each of which had their own technical goals and administrative constraints. RARE (*Réseaux Associés pour la Recherche Européenne*) was created in 1986 as a forum for the European national networking organisations to resolve the problems of interworking between those national services.

The first ideas about the creation of a pan-European service provider offering international services appeared in the late 1980s. It was not however until 1991 that a task force was established under the auspices of RARE. The first blueprint for such an organisation was established quite quickly and by December 1991 a specific proposal had been produced for the creation of a limited liability company with 14 European National Research Networks as shareholders. In February 1993 the company DANTE was established; the acronym stands for *Delivery of Advanced Network Technology to Europe*. It was formally launched and took on its first employees on 5 July 1993.

DANTE company structure

The choice of a limited liability company was made for both legal and financial reasons. As the sums of money involved in providing international network services are significant it is important to have a commercial arrangement with the correct financial controls and limitation of risk. A limited liability company automatically limits risk and allows the control of expenditure to be related directly to the shareholders obligation to pay. Company control is exercised by shareholders' weighted voting, a reflection of shareholding size. Shareholdings fall into four categories, and are related to country size as measured by GNP.

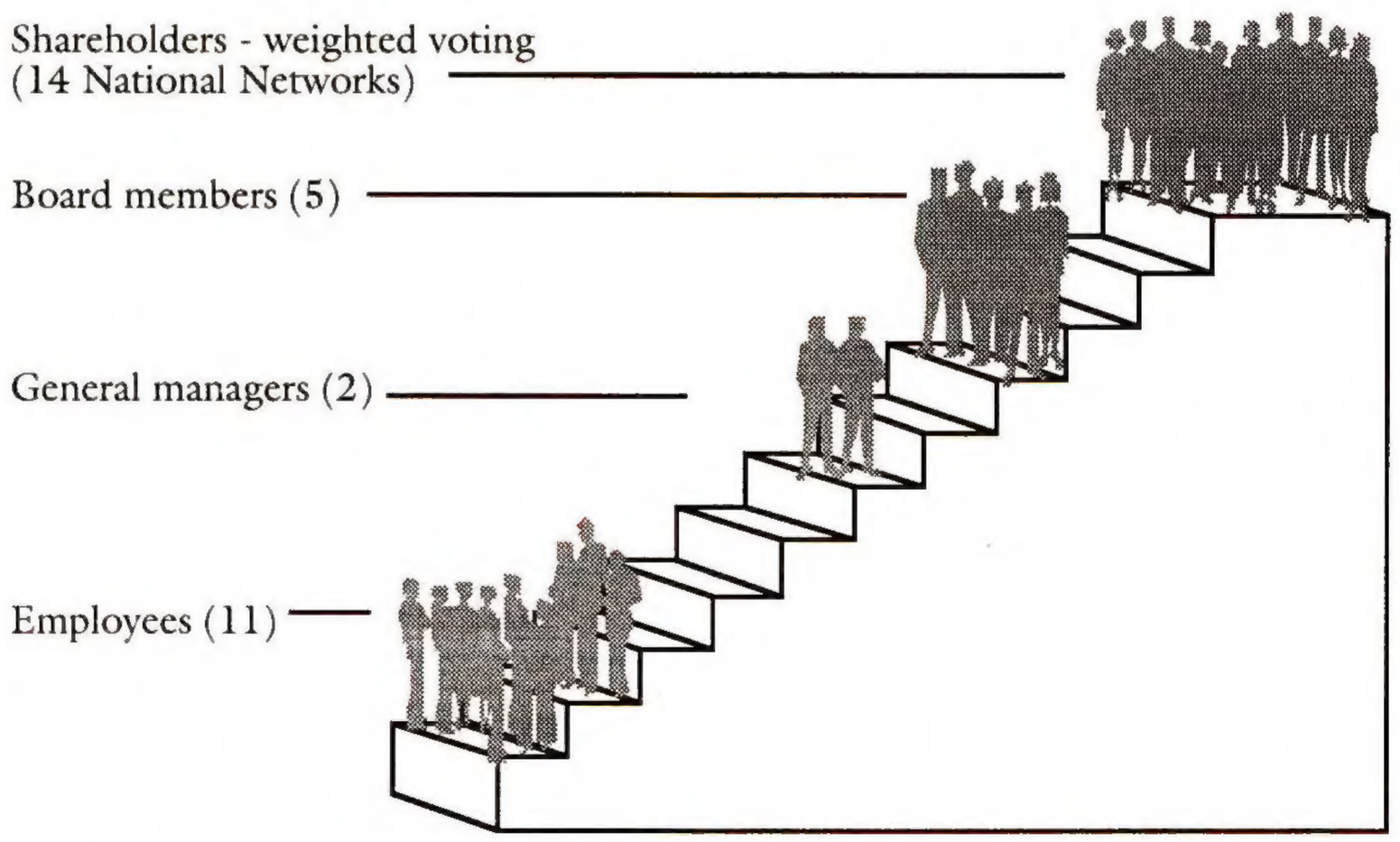


Figure 1: DANTE Company Structure

Figure 1 illustrates the company's structure. The shareholders are represented by a board of five members. The two joint general managers were appointed in the Summer of 1993; 9 other staff have been recruited since, covering the following areas: Network Planning and Development (2), Network Management and Customer Liaison (2), Applications Planning and Development (2), Customer and Public Relations (1) and General Support and Administration (2).

The gestation of EuropaNET

The X.25 part of EuropaNET originated in the IXI (*International X.25 Infrastructure*) Pilot network, which provided connectivity at 64 Kbps between countries participating in the CEC COSINE Project (concluded in April 1993) from July 1990 onwards. COSINE, constrained by its internal organisational procedures and technical orientation towards OSI protocols, was overtaken to some extent by the explosion of usage of TCP/IP in Europe, the result of which was the setting up of the Ebone IP backbone in 1992.

In the specification and tender for a 2Mbps production service to supersede IXI, it had already been determined to be essential that the new backbone should handle multiple protocols. The result of the tender was the setting up in October 1992 of the *European Multi-Protocol Backbone* (EMPB) as the pan-European component of EuropaNET which offered a 2Mbps service in all COSINE member states. DANTE's own transatlantic capacity was added to the EuropaNET service at the beginning of 1994.

EuropaNET and Ebone have coexisted since; a comparison of the "styles" of both services is presented in Table 1.

DANTE – EuropaNET	*Ebone*
Managed service, specified in detail and contracted to professional operational suppliers (including the national research networks).	Coordinated service, taking advantage of latest developments; development and operations closely linked.
Quality of Service (availability, performance) defined in specifications and operational contract.	Best efforts—usually very committed—maximum use of capacity given priority over performance for individual user.
Imposition of Management Discipline (labelled bureaucracy by technicians). More orderly (but slower) progress.	Try it and see if it works; if so OK, if not then deal with problem. Rapid adoption of new techniques.
Predictable behaviour, performance dependable (even if not high).	Actual performance unpredictable, depends on load imposed by others; priorities determined by technicians rather than users.

Table 1: DANTE/EuropaNET versus Ebone organisation and service characteristics

Services

Today EuropaNET offers an international connectivity package consisting of three components:

1. *A pan-European backbone linking the European research networks:*

EuropaNET's main technical and/or value adding features are:

- IP–IP with the external routing protocols EGP and BGP3 (BGP4 is about to be tested)
- CLNP–CLNP with Static Routing and the Inter-Domain Routing Protocol IDRP
- X.25/X.75 interconnections

continued on next page

DANTE and EuropaNET *(continued)*

- IP–X.25/X.75 interconnections
- CLNP–X.25/X75 interconnections
- IP/CLNP policy based routing
- Address authentication & accounting for IP, CLNP and X.25/ X.75
- High-quality network management and help desk service available 24 hours per day, 7 days per week, with fault handling and escalation in accordance with agreed procedures
- Service Level Agreements:
  - Network service availability guarantee, including availability of access lines;
  - Guaranteed end-to-end throughput and delays

The EuropaNET backbone nodes are normally installed in each country where there is a point of attachment. At each node location there may be one or more nodes. Each node is connected to at least two other nodes and each node location is connected to at least two other node locations, up to now in two different countries. Nodes are present in nearly all western-European countries and in a growing number of Eastern European countries. DANTE is cooperating with the CEC's PHARE program to extend EuropaNET eastwards.

The main building blocks in the node equipment are the INMOS Transputer from SGS Thomson Microelectronics and the XPC controller from Motorola. The Transputer provides true parallel processing with very low switching delays and high switching capacity. With the current version of the Transputer and XPC controller, transit delays are 0.7 msecs, access delays are 4 msecs and nodes can be built up to switch several hundred thousand packets per second.

The system is capable of handling trunk line speeds of 8Mbps with full utilisation of the bandwidth. At the end of 1992 a new version of the INMOS Transputer, the T9000, was released. With an appropriate line interface the T9000 is capable of handling 34Mbps trunk and 8Mbps access lines.

2. *A gateway to other European countries and services:*

For a transitional period, access to a further group of European countries and services—not yet connected to EuropaNET—is available via a gateway to Ebone. The gateway is located in Amsterdam and operates at 2Mbps.

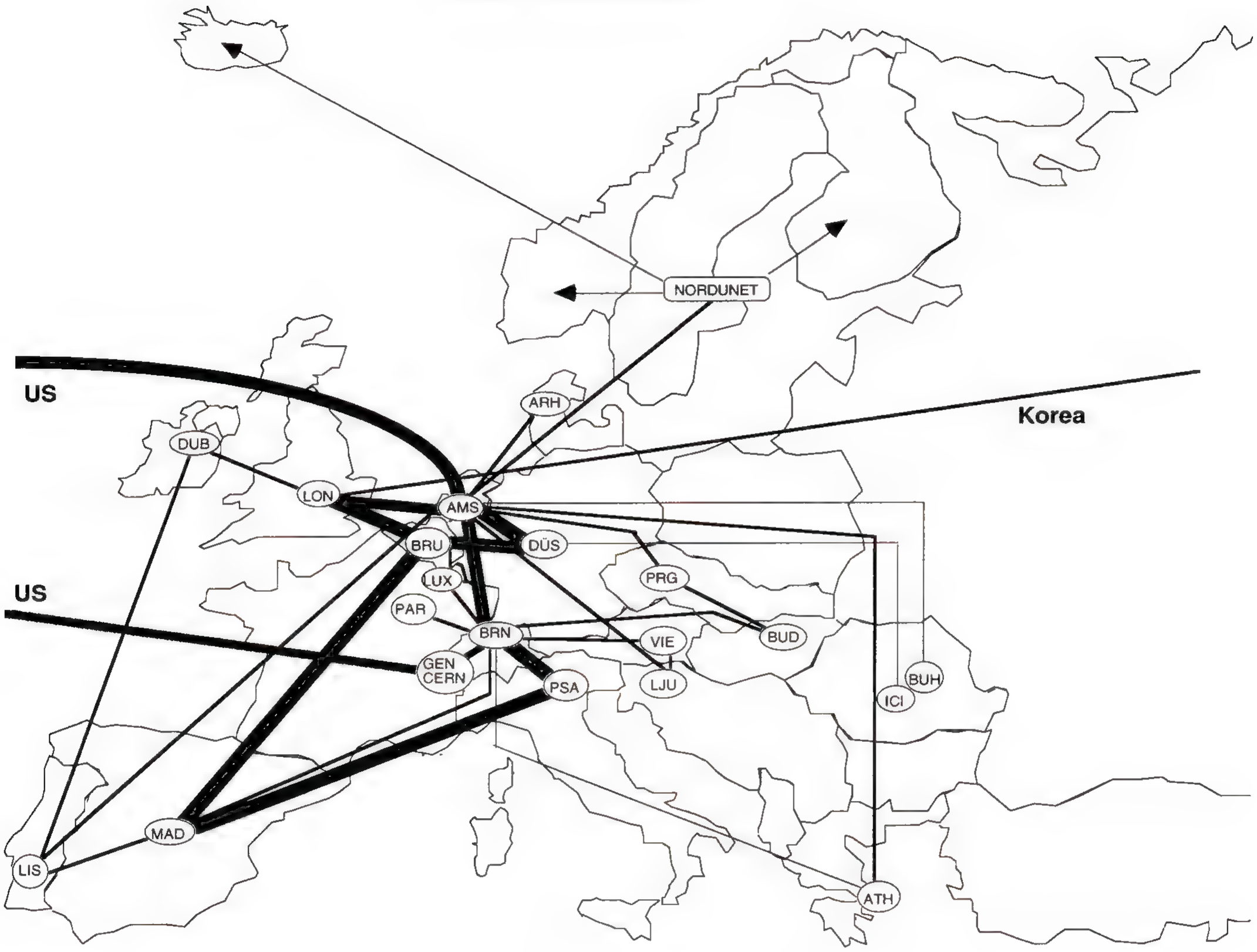
3. *Connections from Europe to other continents:*

Transatlantic connections have historically been provided on a bilateral basis between the US and individual European countries. In order to cater for the needs of those countries which do not have their own intercontinental circuits, DANTE has provided two intercontinental circuits to the USA, an E1 (2Mbps) circuit between Amsterdam and Washington and a T1 (1.5Mbps) circuit between Geneva and Washington.

Lines to Germany, Italy and the UK are currently managed independently of EuropaNET but the national networks concerned have agreed to integrate them with the EuropaNET service for routing and backup purposes.

DANTE
EuropaNET*

Lines ordered or planned, January 1994
*EuropaNET is trademark of DANTE



EuropaNET (Backbone) Access Points

Country	Network	Capacity
Austria	**ACONET**	64 kbps
Belgium	**BELNET**	2 Mbps
	CEC	64 kbps
	DCS	64 kbps
	JRC-GEEL	64 kbps
	RESULB	64 kbps
Czech Rep.	**CESNET**	64 kbps
Denmark	DATAPAK	64 kbps
France	TRANSPAC	64 kbps
Germany	**WIN**	2 Mbps
Greece	**ARIADNET**	64 kbps
	HELLASPAC	64 kbps
Hungary	**HUNGARNET**	64 kbps
	TUB	64 kbps
	PLEASE	64 kbps
Ireland	**HEANET**	64 kbps
	EIRPAC	64 kbps
Italy	**GARR**	2 Mbps
	JRC/ISPRA	64 kbps
	ITAPAC	9.6 kbps
Luxembourg	**RESTENA**	64 kbps
	LUXPAC	64 kbps
Netherlands	**SURFNET**	2 Mbps
	DN1	64 kbps
	ESAPAC	64 kbps
Portugal	**RCCN**	64 kbps
	TELEPAC	64 kbps
Romania	ICI	9.6 kbps
	PUB	9.6 kbps
Slovenia	**ARNES**	128 kbps
	SIPAX.25	64 kbps
Spain	**REDIRIS**	2 Mbps
	IBERPAC	64 kbps
Sweden	**NORDUNET**	64 kbps
Switzerland	**SWITCH**	2 Mbps
	CERN	1 Mbps
UK	**JANET**	2 Mbps

Gateway EuropaNET-Ebone

Amsterdam	2 Mbps

Intercontinental connectivity

Amsterdam - Washington	2 Mbps
Geneva/CERN - Washington	1.5 Mbps
London - Korea	64 kbps

Figure 2: EuropaNET topology in January 1994

continued on next page

DANTE and EuropaNET *(continued)*

NORDUnet has indicated its intention of doing the same with the US–Stockholm line when it switches from Ebone to EMPB in June 1994. DANTE is currently discussing the opportunities for rationalising intercontinental connectivity with the other European circuit operators with the aim of increasing total capacity in a cost effective way.

In the course of 1994 DANTE will also be providing intercontinental connectivity to Korea—a contract for this was signed in January between DANTE and the CEC—and Canada. It is reviewing the possibility of a new direct connection between Europe and Japan.

A next generation backbone

The establishment of EuropaNET was a milestone in the development of European research networking. However, a network access capacity of 2Mbps is still only the beginning, in particular when European national networks are starting to operate national services at 34Mbps and higher speeds. The introduction of these services will create a demand for complementary international facilities. DANTE has taken up the challenge to define and procure a 34Mbps and 155Mbps backbone for the European research community.

A challenge indeed, taking into account the political and technical setting in Europe. A major outstanding issue for the development of European research networking remains the lack of a "central" funding and support organisation (equivalent to the Federal Government in the US) whose responsibilities cover the whole of Europe. Without such an organisation to take initiatives and to seed the creation of a new infrastructure, the collection of necessary funds is a significant management challenge.

The setting up of a high speed backbone represents an opportunity to extend the capacity of the existing backbone. In addition, new developments in the area of network technology and in particular the deployment of *Asynchronous Transfer Mode* (ATM) offer the possibility for European researchers to gain early experience of the benefits and new applications which ATM will enable. Other important issues to be tackled are topology planning, connectivity requirements to other continents, options with respect to management, operation and payment of the service, and an assessment of relevant applications.

DANTE applications

In addition to providing the pan-European backbone network service DANTE organises a range of Value Added Applications. Foremost among these is the X.400 Mail Coordination Service, *MailFLOW*. The Service was established to ensure the efficient interworking of the X.400 e-mail services provided by the national research networks.

MailFLOW coordinates activities between and among the national networks and offers a single information and contact point for the international MHS community. The MailFLOW team, at the Swiss national network SWITCH, maintains an information server with operational documentation such as routing tables and mapping tables, provides trouble-ticket and monitoring functions, supports new MHS services and organises communication between MHS managers, both through e-mail and meetings.

As well as mail coordination, plans are well advanced to build on the pilot directory service PARADISE set up as part of the COSINE Project. DANTE is preparing to offer a coordinated international directory service from May 1994.

DANTE's other important area of activity in the field of Value Added Applications is Information Services. This is a particularly challenging issue. More than any other area, information services are regarded as "free." Neither users nor funding bodies are eager to pay charges that DANTE, with its commercial structure, is obliged to make for all its services. In practice the costs associated with operation of an Information Service platform are significant as is the data management overhead. As a consequence, quality of service and topicality of data are very variable. DANTE is committed to offer cost-based, unbundled prices and to avoid cross subsidy. Information services are more than just a simple technical challenge within this commercial environment.

Liaison Desk

The requirement of a liaison desk in support of the international service package was part of the blueprint for the setting up of DANTE. A particular need was identified for centralising support for EuropaNET. DANTE's liaison desk, *DANTEAM,* started operating in the Spring of 1994.

DANTEAM will act as liaison between Unisource—the company operating the EuropaNET backbone—and staff at the operational departments of the national networks. A trouble-ticket system will be used to register and contribute to the resolution of reported problems. As DANTE's involvement with network development and management gradually grows, operational and administrative responsibilities will increase as well. The liaison desk will play a central role in implementing and coordinating this process. Despite its predominant technical orientation, the liaison desk will also be a first point of contact with regard to Application Services.

Conclusion

After seven years of coordination efforts in European research networking DANTE has appeared on the stage to look after the international networking needs of the European research community. The company has been successful in the first half year of its existence and a firm basis now exists from which new services can be developed. While RARE will continue to coordinate technical discussions of development needs, DANTE organises, purchases and manages services on behalf of its customers, the national research networks in Europe.

Acknowledgement

A number of people in DANTE have provided input for this article, in particular joint General Managers Howard Davies and Dai Davies.

References

[1] Stockman, B., "Current Status on Networking in Europe," *ConneXions,* Volume 5, No. 7, July 1991.

[2] Stockman, B., "Global Connectivity: The Global Internet Exchange (GIX)" *ConneXions,* Volume 7, No. 11, November 1993.

[3] *ConneXions,* Volume 7, No. 5, May 1993, "Special Issue: Focus on Europe."

[4] Stockman, B., "EBONE, The European Internet Backbone," *ConneXions,* Volume 7, No. 5, May 1993.

[5] "Réseaux Associés pour la Recherche Européenne (RARE)," *ConneXions,* Volume 6, No. 1, January 1992.

[6] Onions, J., "Components of OSI: The X.400 Message Handling System," *ConneXions,* Volume 3, No. 5, May 1989.

JOSEFIEN BERSEE witnessed the setting up of the Operational Unit/DANTE during the years she worked for RARE as a Publicity Officer. She has been employed by DANTE as Customer/Public Relations Manager since October 1993. She can be reached via Internet e-mail: `j.bersee@dante.org.uk`

A Brief Comparison of Public Domain SLIP/PPP Drivers for MS-DOS

by Billy Barron, University of Texas at Dallas

Introduction

The University of Texas at Dallas (UTD) is considering installing SLIP and PPP down the road. In preparation for this, I decided to do some testing before any of our equipment was installed. After some tests, I definitely decided that all SLIP, CSLIP (Compressed SLIP), and PPP packages for MS-DOS machines were not created equal.

My first requirement was that the package had to be public domain so we could distribute it free to our users. My second requirement was that it needed to provide a packet driver or ODI interface to the higher level applications.

The test environment

On the PC end, I have a 386 25Mhz with 64Kb cache and 1 MB of memory. My modem was a MultiTech MultiModem II and I kept the configuration of the modem with faculty settings because I figured that most users out there would do the same. The MultiModem II is a V.32/V.32bis/V.42/V.42bis.

On the remote end, I used the facilities of Texas Metronet, who I owe thanks to for helping me perform some of these tests. On their end, I was going through a Livingston PortMaster Terminal Server to reach an HP 705 known as "feenix" which served as the SLIP, CSLIP and PPP host. Feenix is a workhorse running interactive users and acting as a news server. This may have caused some variance in my tests.

Finally, I used my workstation on the UTD campus called "frog" as my connection point for FTP, *Telnet* and *ping*. I used the initial Rutgers version of CUTCP for FTP and *Telnet*. The package has two ways to FTP and the TN3270 program fails back to VT100 when appropriate. I utilized the FTP server built into TN3270 as well as TN3270 for my *Telnet* connections. The *ping* was part of the WATTCP package. Frog is a SuperWorkstation 50Mhz SuperStation II clone. The connection between Metronet and Frog is at Ethernet speeds.

Packages evaluated

I will start by giving some background on the packages I attempted to use.

- *SLIP8250:* I found this SLIP driver in the Crynwr packet driver distribution.
- *EtherSLIP:* A modified SLIP8250 that supports Ethernet packets.
- *Slipper:* Slipper was written by Peter Tattum who is also responsible for the newsreading package *Trumpet*.
- *Cslipper:* A compressed SLIP version of *Slipper*.
- *UMSLIP:* The University of Minnesota SLIP package.
- *EtherPPP:* A PPP driver developed by University of Michigan and Merit.
- *Kermit:* This is only included for comparison purposes.

In all cases, I spent some test trying to optimize the particular package.

FTP performance tests

The purpose of the FTP test was to show the raw data transfer rate for the various drivers. The results should also be useful in estimating performance of tools like *Mosaic*.

In this test, I downloaded three files. The first was a random *PostScript* document called **Save.ps.** The second was a compressed ZIP 2.0 file. The final was the ASCII MS-Kermit documentation. I downloaded the files three times and averaged the results.

In theory, we should see that Compressed SLIP performs the best since it compresses TCP/IP headers. Second should be PPP because it does the same compression, but it adds an additional three bytes, which should make a very minor difference in the numbers. In last place we should find SLIP.

	SLIP8250 .6	Slipper EtherSLIP 1.4	UMSLIP 1.4	UMSLIP 2.0
Save.ps 55,811b ASCII	Too Slow	41.54s 1.32K/s	40.94s 1.36K/s	Crashed	41.81s 1.33K/s
zman01.zip 767,529b BINARY	Too Slow	559.23s 1.34K/s	503.90s 1.49K/s	Crashed	569.03s 1.31K/s
ckuker.doc 251,390b ASCII	Too Slow	194.51s 1.29K/s	170.24s 1.48K/s	Crashed	189.61s 1.33K/s

	Cslipper 1.4	EtherPPP 1.9.37beta	Kermit
Save.ps 55,811b ASCII	33.09s 1.67K/s	30.26s 1.80K/s	39.30s 1.38K/s
zman01.zip 767,529b BINARY	475.64s 1.57K/s	487.91s 1.54K/s	851.86s 0.88K/s
ckuker.doc 251,390b ASCII	142.67s 1.76K/s	134.62s 1.85K/s	169.97s 1.44K/s

Figure 1: FTP Performance Results

To start with SLIP8250 with CUTCP was transferring the file too slow for me to complete the tests in a reasonable amount of time. It was my best guess that would have taken hours to complete the transfer. The other problem was UMSLIP 1.4 which crashed the connection during the tests every single time.

In the SLIP category, Slipper won hands down against EtherSLIP and UMSLIP 2.0. As expected, Cslipper beat all SLIP drivers. EtherPPP unexpectedly beat Cslipper in two out of three of the files.

continued on next page

Comparison of SLIP/PPP Drivers *(continued)*

Ping tests

The purpose of the *ping* test was to measure interactive responsiveness. From Telnet, it was nearly impossible to really say if one driver was more responsive than another.

The *ping* tests turned out to not be very useful. All of the packages except one had *ping* times of .10 or .11 seconds. That package was EtherPPP which had a disappointing .26 second result. Unfortunately, I have not been able to figure out why EtherPPP was slower. It should be noted that EtherPPP seemed no slower when using *Telnet* than any of the other drivers.

Conclusions

After the tests, I decided that really only three of the packages are worthy of further consideration. First, if you are in a situation where you only have SLIP access, then use Slipper. Otherwise use either Cslipper or EtherPPP depending on your needs. However, you probably should run your own performance evaluation especially since some of the drivers are still being refined.

References

[1] C. Partridge, "Dialup IP," *ConneXions,* Volume 3, No 11, November 1989.

[2] J. Romkey, "A Nonstandard for Transmission of IP Datagrams over Serial Lines: SLIP," RFC 1005, June 1988.

[3] D. Perkins, "The Point to Point Protocol: A proposal for Multiprotocol Transmission of Datagrams over Point to Point Lines," RFC 1134, 1989.

[4] D. Perkins & R. Hobby, "The Point to Point Protocol Initial Configuration Options," RFC 1172, July 1990.

[5] R. Hobby, "The Point to Point Protocol (PPP)—A new proposed standard Serial Line Protocol," *ConneXions,* Volume 4, No. 4, April 1990.

[6] J. Romkey, "SLIP: Serial Line IP," *ConneXions,* Volume 2, No. 5, May 1988.

[7] J. Romkey, "The Packet Driver," *ConneXions,* Volume 4, No. 7, July 1990.

[8] R. Coop, "SLIP Interoperability," *ConneXions,* Volume 6, No. 6, June 1992.

[9] R. Coop & B. Tompsett, "An Investigation of SLIP and Dialup SLIP," Research Report 91/3, Department of Computer Science, University of Hull, November 1991.

[10] R. Coop & B. Tompsett, "Current Commercial and Public Implementations of SLIP, Dialup SLIP and PPP for SunOS UNIX Systems and IBM Compatible PCs," Report for the UK Internet Consortium, Department of Computer Science, University of Hull, November 1991.

[11] Finseth, C., "SLIP at the University of Minnesota," *ConneXions,* Volume 7, No. 1, January 1993.

BILLY BARRON has an M.S. in Computer Science from the University of North Texas. He also worked there for many years until he recently transferred to the University of Texas at Dallas. In parallel, he has worked on the CICNet Electronic Journal Archive, but recently resigned to have more free time. Also, he has the co-author of a famous guide to libraries on the Internet. E-mail: `billy@utdallas.edu`

NetCash: Electronic Currency for the Internet

**by Gennady Medvinsky and B. Clifford Neuman,
University of Southern California**

Abstract

NetCash is a framework for electronic currency being developed at the Information Sciences Institute of the University of Southern California. NetCash will enable new types of services on the Internet by providing a real-time electronic payment system that satisfies the diverse requirements of service providers and their users. Among the properties of the NetCash framework are: security, anonymity, scalability, acceptability, and interoperability.

One of the primary goals of NetCash is to facilitate anonymous electronic payments over an unsecure network without requiring the use of tamper-proof hardware. NetCash is designed to provide secure transactions in an environment where attempts at illegal creation, copying, and reuse of electronic currency are likely. In order to protect the privacy of parties to a transaction, NetCash provides financial instruments that prevent traceability and preserve the anonymity of users.

Furthermore, with NetCash, service providers and their users are able to select payment mechanisms based on the level of anonymity desired, ranging from non-anonymous and weakly anonymous instruments that are scalable, to unconditionally anonymous instruments that require more resources of the currency server. NetCash provides scalable electronic currency that is accepted across multiple administrative domains.

Using the NetCash framework, parties that are customers of different banks can accept each other's currency. To provide interoperability across currency servers, NetCash integrates anonymous electronic currency into the non-anonymous electronic banking infrastructure that has been proposed for routine transactions. This article presents an overview of NetCash; a more detailed description can be found in [3].

Requirements

Among the desirable properties for an electronic currency system are: security, anonymity, scalability, acceptability, and interoperability.

- *Security:* Forging paper currency is difficult. Unfortunately, electronic currency is just data and is easily copied. Copying or double spending of electronic currency should be prevented or detected. Ideally the illegal creation, copying, and reuse of electronic cash should be unconditionally or computationally impossible. Some systems rely instead on post-fact detection and punishment of double spending [1].

- *Anonymity:* The identity of an individual using electronic currency should be protected; it should not be possible to monitor an individual's spending patterns, nor determine one's source of income. An individual is traceable in traditional transaction systems such as checks and credit cards. Some protocols are unconditionally untraceable, where an individual's spending cannot be determined even if all parties collude [1]. For some transactions, weaker forms of anonymity may be appropriate, e.g., traceability can be made difficult enough that the cost of obtaining such information outweighs the benefit.

continued on next page

Electronic Currency for the Internet *(continued)*

- *Scalability:* A system is scalable if it can handle the addition of users and resources without suffering a noticeable loss of performance. The existence of a central server through which transactions must be processed limits the scale of the system. The mechanisms used to detect double spending also affect scalability. Most proposed e-cash protocols assume that the currency server will record all coins that have been previously spent and check this list when verifying a transaction [1,5,6]. This database will grow over time, increasing the cost to detect double spending. Even if the life of a coin is bounded, there is no upper bound on the amount of storage required since the storage requirement depends on the rate at which coins are used, rather than on the number of coins in circulation.

- *Acceptability:* Most e-cash proposals use a single bank [1,5,6]. In practice, multiple banks are needed for scalability, and because not all users will be customers of a single bank. In such an environment, it is important that currency minted by one bank be accepted by others. Without such acceptability, electronic currency could only be used between parties that share a common bank. When currency minted by one bank is accepted by others, reconciliation between banks should occur automatically. To our knowledge, NetCash is the first system that satisfies this requirement.

- *Interoperability:* Users of the Internet will select financial instruments that best suit their needs for a given transaction. It is likely that several forms of electronic currency will emerge, providing different tradeoffs for security, anonymity, and scalability. In such an environment it is important that funds represented by one mechanism be easily convertible into funds represented by others.

Framework

The NetCash infrastructure is based on independently managed, distributed currency servers that provide a point of exchange between anonymous electronic currency and non-anonymous instruments such as electronic checks. In the framework, checks based on the global accounting infrastructure [4] support the transfer of funds between currency servers, forming a financial federation where currency minted by different servers is accepted. An organization wishing to set up and manage a currency server obtains insurance for the new currency from an agency similar to the United States *Federal Deposit Insurance Corporation* (FDIC); the currency is backed by account balances registered to the currency server in the non-anonymous accounting infrastructure. A certificate of insurance allows the coins minted by a currency server to be accepted across administrative domains.

Figure 1 shows a financial federation consisting of several accounting servers (AS1, AS2 and AS3) and several currency servers (CS1 and CS2). The accounting servers maintain accounts for the currency servers and other clients. Funds can be transferred between currency servers when one (the payor) issues a check authorizing another (the payee) to transfer funds from the payor's account to that of the payee. The accounting hierarchy allows the clearing of such checks between the accounting servers that maintain the accounts for the respective currency servers. Although the currency servers are identified in such transactions, the end-users are not.

Such transfers provide for anonymous currency transactions between users in different administrative domains. To verify a coin issued by another currency server, a user's local currency server contacts the remote currency server to convert the coin, accepting in return a check payable to the local currency server. This check is then cleared through the global accounting infrastructure, and a new coin issued by the local currency server is returned to the user.

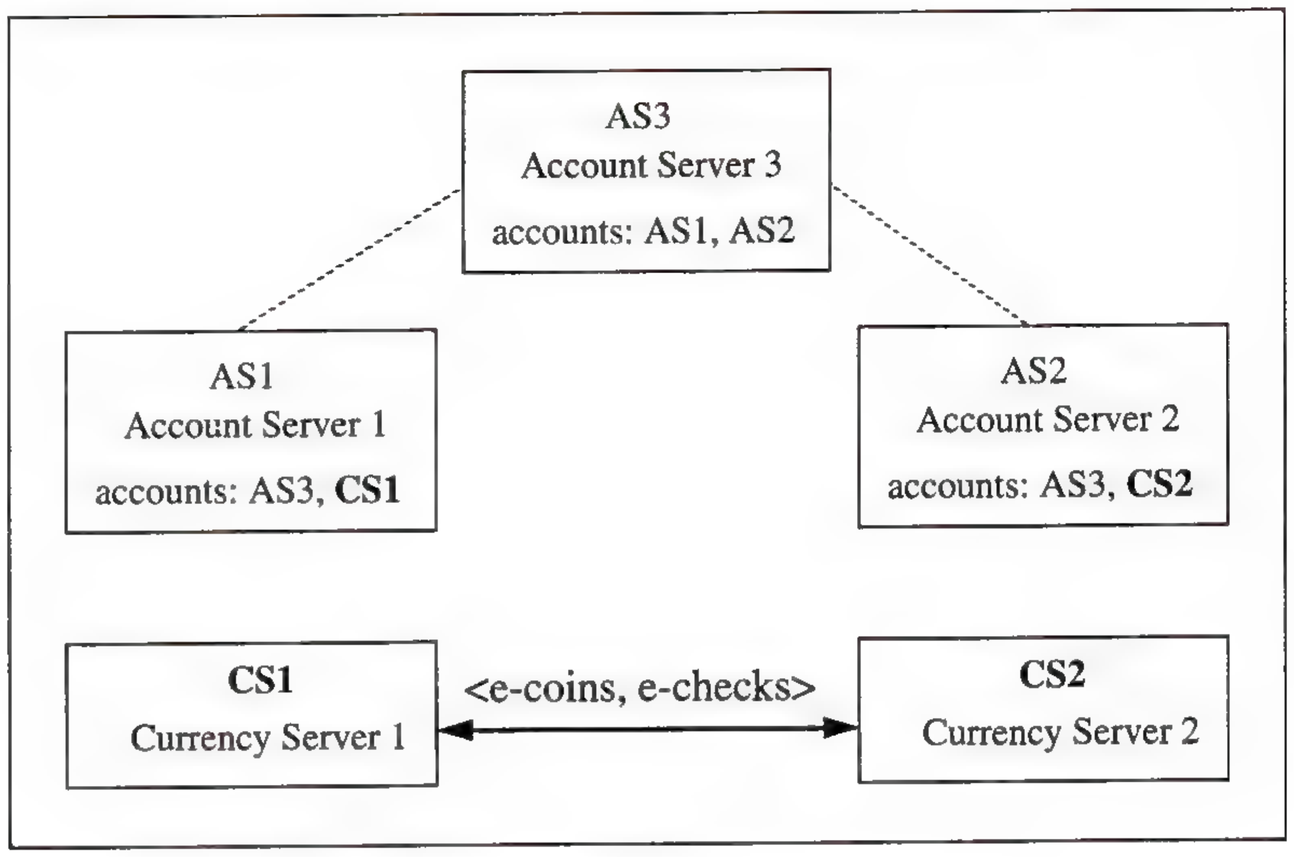


Figure 1: Financial federation

Electronic currency transaction

NetCash provides a framework for integrating currency servers using different electronic currency protocols, providing a range of anonymity guarantees. Electronic currency mechanisms providing unconditional or weak anonymity can be tied to the framework. Either the payor, the payee, or both parties may remain anonymous. The integration of multiple mechanisms allows a tradeoff between the anonymity guarantees and the resulting overhead of the electronic currency mechanism. Economic incentives can be used to encourage the selection of an appropriate mechanism.

Chaum and others have proposed protocols supporting unconditionally untraceable electronic currency [1]. While those protocols may be used within the framework we are developing, we have chosen to concentrate our efforts on protocols that provide weaker anonymity guarantees. The protocols we are developing are described briefly here; additional detail is available in [3].

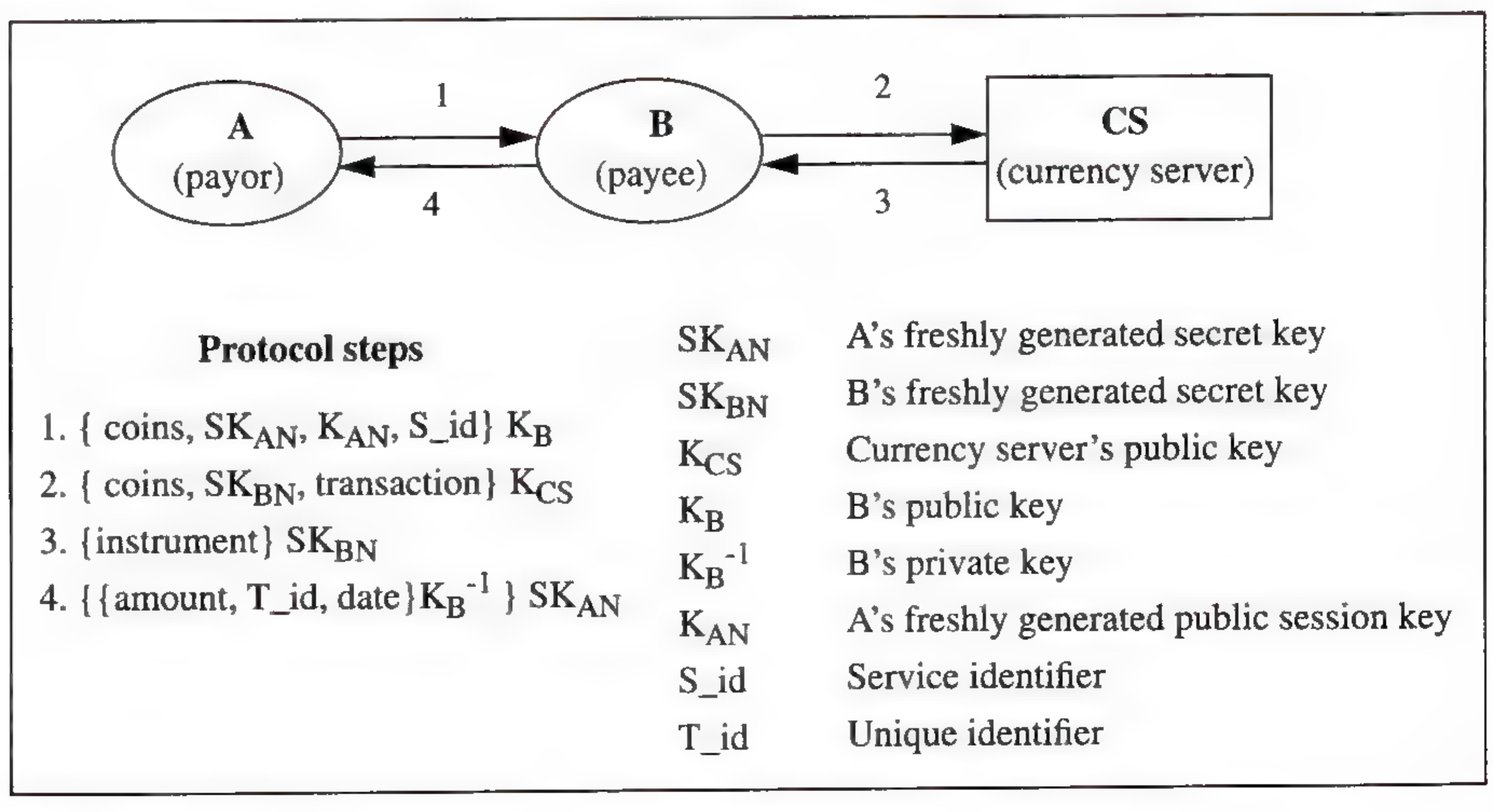


Figure 2: Protocol for routine monetary transactions using NetCash

continued on next page

Electronic Currency for the Internet *(continued)*

Figure 2 on the previous page shows a protocol for routine monetary transactions using NetCash. A payment is made by *A* (the payor) to *B* (the payee). *A* remains anonymous and *B* is protected from fraud. It is assumed that *A* can determine *B*'s public key and *B* can determine the public key of the currency server. In the first step, *A* sends coins, the identifier of the desired service *S_id,* and two keys; a freshly generated secret key SK_{AN} and a public session key K_{AN}, all encrypted in *B*'s public key. *B* records the newly chosen secret key SK_{AN} which will be used to establish a secure channel with *A* at a later time. A public session key can also be used to verify that subsequent requests originate from the principal that paid for the service.

After receiving coins from *A, B* verifies the validity of the coins with the currency server. In the second step, *B* sends to the currency server the coins, SK_{BN} a newly chosen secret key, and an indication of what it wants back: new coins known to be valid, or a check made payable to a named principal. This message is encrypted in the currency servers public key. Upon receiving the coins, the currency server verifies the validity of the coins. If the coins haven't been spent already, the server returns the desired instrument sealed with SK_{BN}. In the final step, *B* returns a receipt signed with its private key and encrypted with SK_{AN}. The receipt includes amount paid, date and a unique identifier *T_id* that will be used along with the session key to obtain the service.

One shortcoming of this protocol as described is that it provides protection from fraud only by the payor. *B* can spend *A's* coins without providing a valid receipt. Extensions to the protocol supporting protection against fraud for both parties, anonymity provisions for the payee, and partially offline transactions are described by the authors in [3].

Projects status

Several security projects are under way at ISI that provide the necessary infrastructure for NetCash. One project is developing a proxy mechanism using the *Kerberos* authentication system [7]. This mechanism provides the authorization services needed for the non-anonymous accounting infrastructure that connects currency servers. Work is also under way to develop the accounting databases that are needed for the non-anonymous accounting infrastructure and the currency databases needed by electronic currency servers.

References

[1] Chaum, D., Fiat, A., Naor, N., "Untraceable Electronic Cash," in Proceedings of Crypto '88, 1988.

[2] Even, S., Goldreich, O., Yacobi, Y., "Electronic Wallet," in Proceedings of Crypto '83, 1983.

[3] Medvinsky, G., Neuman, B. C., "NetCash: A Design for Practical Electronic Eurrency on the Internet," in: Proceedings of the First ACM Conference on Computer and Communications Security, November 1993.

[4] Neuman, B. C., "Proxy-based Authorization and Accounting for Distributed Systems," in: Proceedings of the 13th International Conference on Distributed Computing Systems, May 1993.

[5] Okomato, T., Ohta, K., "Universal Electronic Cash," in Proceedings of Crypto '91, 1991.

[6] Pfitzmann, B., Waidner, M., "How to Break and Repair a 'Provably Secure' Untraceable Payment System," in Proceedings of Crypto '91, 1991.

[7] Steiner, J. G., Neuman, B. C., Schiller J. I., "Kerberos—An Authentication Service for Open Network Systems," in: Proceedings of the Winter 1988 USENIX Conference, February 1988, pp 191–201.

[8] Schiller, J., "Kerberos: Network Authentication, *ConneXions,* Volume 4, No. 1, January 1990.

[9] *ConneXions,* Volume 4, No. 8, August 1990, "Special Issue on Network Management and Network Security."

[10] W. Diffie, "The first ten years of public-key cryptography," *Proceedings of the IEEE,* 76(5):560–577, May 1988.

[11] W. Diffie & M.E. Hellman, "Privacy and authentication: An introduction to cryptography," *Proceedings of the IEEE,* 67(3):397–427, March 1979.

[12] R. L. Rivest, A. Shamir, & L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM,* 21(2):120–126, February 1978.

[13] W. Diffie & M.E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory,* IT-22:644–654, 1976.

[14] Ronald L. Rivest, "Cryptography," In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science,* Volume 1, pages 719–755, Elsevier Science, 1990.

[15] Kaliski, B., "An Overview of Public-Key Cryptography Standards," *ConneXions,* Volume 6, No. 5, May 1992.

The work described in this article was supported in part by the Advanced Research Projects Agency under NASA Cooperative Agreement NCC-2-539. The views and conclusions presented should not be interpreted as representing the official policies of the funding agencies.

[Ed.: This article is reprinted with permission from *EM—Electronic Markets,* Issue No. 9/10 1993, Institute for Information Management, University of St. Gallen, CH-9000 Switzerland. For subscription information contact: `zbornik@sgcl1.unisg.ch`].

GENNADY MEDVINSKY is a research assistant and Ph.D. candidate in Computer Science at the University of Southern California (USC). He received a B.A. from the University of California San Diego (1990) and holds a Masters degree from USC (1993) in Computer Science. His primary research interests include distributed systems and computer security. Gennady is working on authorization, accounting, and payment infrastructure for distributed systems as part of his dissertation. He may be reached as: `ari@isi.edu`.

CLIFFORD NEUMAN is a scientist at the Information Sciences Institute of the University of Southern California (USC). Dr. Neuman has been working on computer security since 1985. After receiving a S.B. degree from the Massachusetts Institute of Technology in 1985 he spent a year working for Project Athena where he was one of the principal designers of the *Kerberos* authentication system. Dr. Neuman received M.S. and Ph.D. degrees from the University of Washington, where he designed the *Prospero* Directory Service which is widely used to locate information from Internet archive sites. Dr. Neuman's recent work in the security area includes the development of a security infrastructure supporting authorization, accounting, and electronic payment mechanisms. He may be reached on the Internet as: `bcn@isi.edu`.

Call for Papers

The *Network Services Conference 1994* (NSC '94) will be held November 28–30 1994 in London, England.

Overview

Open computer networking is no longer the sole domain of universities and research institutions. Today, governments, schools, public organizations, commercial enterprises and private individuals are actively using and supplying information over the global Internet.

How will these various network communities cooperate and interact? How will the academic and research community adapt to the new network reality? How will the network and networking tools now available stand up to the explosion in number of users and amount of information available? How will we train novices? What will we pay for and what will be for free as the commercialization of the network progresses? Will we be inundated by advertising over the net? These are only a few of the questions facing network service providers and users alike.

Building on the success of the previous Network Services Conferences in Pisa (1992) and Warsaw (1993), NSC '94 will focus on the issue of providing services to customers, with special attention paid to the exciting developments in global tools and services. We will address the impact of the new global tools on service development and support, the changing function of traditional tools and services (such as archives), new services (such as multi-media communications), the future role of the library and the effects of commercialization on networks and network services. Customer support at all levels, and the role of support in accessing global services, will also be covered. Talks, tutorials, demonstrations and other conference activities will address the needs of the research, academic, educational, governmental, industrial, and commercial network communities.

NSC '94 is being organized by the EARN (*European Academic and Research Network*) Association in cooperation with the Internet Society, RARE, NORDUnet and EUnet.

Tutorials, demonstrations and posters

In addition to the presentation of papers, there will be tutorial sessions on specific network services as part of the regular conference program. A room will be available for workstations and PCs to be used for demonstrations throughout the conference.

A poster wall will be available to participants for the display of their posters and projects. Terminals with connectivity to the Internet will be available to delegates.

Topics

The Program Committee for NSC '94 is soliciting proposals for papers, tutorials, demonstrations and posters in all fields related to network services. Subject areas for presentations include, but are not limited to, the following:

- Network resource tools
- Network directory services
- Multimedia Communications
- Electronic Publishing
- Libraries and Networking
- Special Interest Communities
- Groupware, Cooperative Work over the Network
- Networking for Schools

- User Support
- Delivering Services to the Desktop
- Quality of Network Services
- Commercialization of Network Services
- Businesses on the Network
- Providing Network Services to New Countries and Communities

Submissions

Papers and proposals for tutorials, demonstrations or posters, including a short biography and an abstract should be sent by mail, fax or preferably by e-mail, to:

NSC '94
EARN Office
PSI – Batiment 211
91405 Orsay CEDEX
FRANCE
Fax: +33 1 6941 6683
E-mail: `NSC94@EARNCC.EARN.NET` or `NSC94@EARNCC.BITNET`

The official language of the conference will be English.

Further information and general inquiry

Further information will be available through the conference mailing list, `NSC94-L@EARNCC.EARN.NET` (or `NSC94-L@EARNCC.BITNET`). If you want to make sure you receive registration information as well as the preliminary program and other information of interest to conference participants, join the list by sending e-mail to:

`LISTSERV@EARNCC.EARN.NET`

or

`LISTSERV@EARNCC.BITNET`

with the line:

`SUB NSC94-L` *Your Name*

Conference information is also available from the EARN anonymous *ftp* server (`ftp.earn.net`) and from the EARN *Gopher* server at `gopher.earn.net`.

If you have any questions or require any assistance, you can contact the conference organizers at:

NSC '94
EARN Office
PSI – Batiment 211
91405 Orsay CEDEX
FRANCE
Tel.: +33 1 6941 2426
Fax: +33 1 6941 6683
E-mail: `NSC94@EARNCC.EARN.NET` or `NSC94@EARNCC.BITNET`

Important dates

Deadline for papers:	July 1, 1994
Deadline for demonstrations and posters:	September 16, 1994
Notification of acceptance (papers/tutorials):	August 1, 1994
Notification of acceptance (posters/demos):	September 30, 1994

Call for Participation

The *Workshop on Mobile Computing Systems and Applications* will be held December 8–9, 1994 at the Dream Inn in Santa Cruz, California. The event is sponsored by the IEEE Computer Society TCOS in cooperation with ACM SIGOPS and the USENIX Association.

Topics

A major challenge of this decade is the effective exploitation of two symbiotic technologies: *portable computers* and *wireless networks*. Harnessing these technologies will dramatically change the computing landscape. But realizing the full potential of the resulting mobile computing systems will require advances in many areas such as:

- Hardware
- Communications
- Scalability
- Power Management
- Security
- Data Access
- User Interfaces
- Location Sensitivity

Format

The goal of this workshop is to foster exchange of ideas in mobile computing among workers in the field. Attendance will be limited to about 60 participants, based on the position papers submitted. Submissions should be fewer than five pages in length and may expose a new problem, advocate a specific solution, or report on actual experience.

In addition, we will be hosting a small number of novel hardware and software exhibits relevant to mobile computing. The exhibits may be research prototypes or commercial products. Interested parties should submit technical descriptions of their exhibits.

Online copies of the position papers will be made available via anonymous FTP prior to the workshop. A printed proceedings will be published after the workshop, and mailed to participants.

A small number of graduate students will be granted a waiver of the registration fee. In return, these students will be required to take notes at the workshop and help put together the proceedings. Students who wish to be considered for the waiver must send in a brief description of their current research, and an explanation of how participation in the workshop is likely to help them.

Submissions

Send (10 x) position papers to:

M. Satyanarayanan
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
Phone: +1 412-268-3743
Fax: +1 412-681-5739
E-mail: `satya@cs.cmu.edu`

Send exhibit descriptions to:

Peter Honeyman
CITI
University of Michigan
Ann Arbor, MI 48103–4943
Phone: +1 313-763-4413
Fax: +1 313-763-4434
E-mail: `honey@citi.umich.edu`

Submissions due:	August 20, 1994
Acceptance Notification:	October 1, 1994
Camera-ready copy due:	November 15, 1994

Call for Papers

Interop Company is soliciting technical papers for an *Engineer's Conference* to be held as part of the upcoming *NetWorld+Interop 94 Paris* event, October 24–28, 1994, in Paris.

The Engineers Conference, which will be held on Thursday/Friday October 27–28, is a two-day focused event offering approaches and solutions to practical systems and software design for networking. All participants in the conference will be able to attend the NetWorld+ Interop 94, Paris exhibition, which will run from October 26–28.

Format

The conference will feature the presentation of original papers which will have been selected by a review committee. All accepted papers will be published in Conference Proceedings. Accepted papers must be presented by original authors during the 2-day conference. Conference sessions typically will be 90 minutes long and will present three papers of 20 to 30 minutes duration.

Topics

The Engineer's Conference will concentrate on engineering design problems in three areas: *High Speed Networking, Internetworking,* and *Multimedia.* The conference seeks to bring together research scholars, engineers, and vendors to address pragmatic engineering issues in the field of networking and distributed systems interoperability. It is an excellent forum for engineers and researchers to publish papers and to be brought up to date on solutions to today's engineering related problems. Papers are solicited in the following areas:

- *High Speed Networking:* ATM, Fast Ethernet, SDH, FDDI-II, HIPPI, SMDS, Frame Relay, Broadband ISDN, etc.
- *Internetworking:* Addressing Schemes, Routing Protocols, Support of Mobility, Design of Bridges, Routers, and Multiprotocol Brouters, Application Gateways etc.
- *Multimedia Networking:* Multimedia technologies, Multimedia interoperability, Packet Video and Voice, Multimedia Mail and Conferencing, Tele-Presence, Virtual Reality, etc.

Submission guidelines

Interested authors are invited to submit an abstract (up to 600 words) clearly describing the problem and the solution offered. Authors of accepted abstracts must submit the paper before the last date. These papers are reviewed by a technical committee for technical merit of the paper before final acceptance. Final camera-ready papers must not exceed 10 A4 pages. All abstracts must contain the authors name, address, telephone number, Fax number and e-mail address (if available). Please send your abstract to:

Interop Europe
14 Place Marie-Jeanne Bassot
92593 Levallois Peret Cedex
Paris
FRANCE
Attention: Engineer's Conference

or e-mail it (ASCII or *PostScript*) to: `Paris_Engineer@interop.com`
(E-mail submission is preferred.)

Important dates

Abstracts due:	June 3, 1994	*(Send it today!)*
Notification to authors:	June 24, 1994	
Draft paper due:	July 22, 1994	
Feedback to authors:	August 5, 1994	
Camera-ready copy due:	September 9, 1994	

CONNEXIONS
303 Vintage Park Drive
Suite 201
Foster City, CA 94404-1138
Phone: 415-578-6900
FAX: 415-525-0194

ADDRESS CORRECTION
REQUESTED

FIRST CLASS MAIL
U.S. POSTAGE
PAID
SAN JOSE, CA
PERMIT NO. 1

CONNEXIONS

EDITOR and PUBLISHER Ole J. Jacobsen

Printed on recycled paper